EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

A Diversity-Preserving Multi-Objective PSO Framework for SFC Scheduling in a Digital Twin-Enabled IIoT Architecture Toward Industry 5.0

Feng Wang[1*], Ning Wu[1]

[1]Anhui Sanlian University, Hefei,China

Abstract

INTRODUCTION: Under Industry 5.0, discrete manufacturing faces a fundamental dilemma: it must balance human-centric flexibility with sustainable and resilient operations. Resolving this dilemma requires the Industrial Internet of Things (IIoT) to coordinate and integrate the three types of heterogeneous resources—sensing, computing, and communication—found in factory workshops, production facilities, and assembly lines. We present an IIoT architecture centered on the digital twin, integrating these capabilities into a single closed-loop feedback chain to resolve resource contention issues at the source, that arise when various manufacturing tasks compete for limited computing power and network bandwidth.
OBJECTIVES: The architecture is physically divided into four layers—end devices, access, edge, and cloud—with each layer coupled via a global feedback channel based on the digital twin, giving the manufacturing environment end-to-end visibility and adaptive resource allocation. For SFC scheduling, we formulate a multi-objective optimization model that simultaneously considers three metrics—total end-to-end task delay, system-wide energy consumption, and inter-node load variance—while imposing hierarchical resource constraints inherent to multi-tier industrial deployments.
METHODS: The algorithm used to solve the above model is ADP-MOPSO, a multi-objective particle swarm optimization method that balances convergence and diversity. It integrates workload-aware heuristic initialization, a dynamic constraint repair operator, crowding-distance-based archive management, and roulette-wheel leader selection within a single PSO framework, whose strategies interact through an externally maintained elite archive.
RESULTS: We tested three IIoT scales based on real-world manufacturing deployment parameters (small, medium, and large; 12/32/64 nodes, 4/8/16 edge servers) and compared against five benchmark algorithms. Across all three scales, ADP-MOPSO achieved the best Hypervolume (HV) and Inverted Generational Distance (IGD) values and remained competitive with the strongest baselines on Spacing (SP). Ablation experiments showed that heuristic initialization contributes a 6.8% HV improvement, and the diversity-preserving module adds a further 4.2%, yielding an overall gain of 10.2%.
CONCLUSION: This study provides an integrated architecture-algorithm solution for real-time multi-objective resource optimization in Industry 5.0 manufacturing systems. Multi-scale simulations calibrated to industrial production settings confirm the effectiveness and scalability of the proposed approach.

Keywords: Industry 5.0, Industrial Internet of Things (IIoT), Service Function Chain (SFC) scheduling, ADP-MOPSO, Digital Twin, Discrete Manufacturing

Received on 06 August 2026, accepted on 09 September 2026, published on 24 September 2026

doi: 10.4108/eetsis.14352

*Corresponding author. Email: rousi1389@126.com

1. Introduction

Discrete manufacturing in the Industry 5.0 era must reconcile human-centric flexibility with sustainable and resilient production. Achieving this goal on the factory floor requires the support of an Industrial Internet of Things (IIoT) infrastructure. This infrastructure must not only provide ultra-low end-to-end latency to meet the demands of real-time process control but also be capable of scaling task orchestration across heterogeneous devices and dynamically adjusting resource allocation in response to fluctuations in production demand.

However, the existing "cloud-edge-endpoint" deployment model has limitations in the following areas: First, the lack of deep integration between sensing, computing, and communication modules leads to multi-hop transmission delays, which are unacceptable for time-sensitive operations. Second, due to the absence of cross-layer coordination mechanisms, the system struggles to strike an effective balance between throughput and load stability.

More fundamentally, conventional architectures operate in an open loop: without a global feedback channel, they cannot adjust resource allocation in response to real-time conditions on the production line. These limitations grow especially severe in discrete manufacturing, where tasks arrive in bursts with diverse computational profiles and must adhere to strict Service Level Agreements (SLAs).

This paper presents a digital-twin-enabled four-tier IIoT architecture (terminal-access-edge-cloud) that unifies sensing, computing, and intelligent decision-making. A digital-twin-enabled global feedback link delivers real-time end-to-end observability and closes the scheduling loop across all four tiers—a capability missing from existing cloud-edge frameworks. Based on this architecture, a multi-objective optimization model is formulated for Service Function Chain (SFC) scheduling under hierarchical resource constraints. We further propose an Archive-Based Diversity-Preserving Multi-Objective Particle Swarm Optimization (ADP-MOPSO) algorithm. To the authors' knowledge, ADP-MOPSO is the first PSO variant to incorporate heuristic initialization, dynamic constraint repair, crowding-distance-based archive screening, and roulette wheel leader selection within a unified framework for industrial SFC scheduling. Multi-scale simulations calibrated to factory deployment parameters demonstrate the effectiveness and scalability of the method, supporting intelligent resource orchestration for Industry 5.0 manufacturing.

2. Related work

2.1. Research on Integrated Sensing, Computing, and Intelligence Architectures

Work on integrated sensing, computing, and intelligence architectures has branched into theory, algorithms, and deployed systems. Polese et al. [1] argued that 6G networks will converge around AI-native designs that bring sensing together with communication and computing in a single stack, with open RAN interfaces and virtualization supplying the real-time data path and on-demand inference. Wymeersch et al. [2] extended this reasoning at the cross-layer level, combining multi-band links, massive MIMO, and RIS into an ISAC framework that treats 6G as a programmable substrate for autonomous mobility and digital twins. Gkonis et al. [3] surveyed the use of AI and machine learning for resource optimization and threat detection in industrial settings, reviewed open-access protocols and their security requirements, and concluded that any practical 6G architecture must weave multiple technologies together. Naeem et al. [4] studied how RIS can strengthen physical-layer security and protect user privacy, while cataloguing the new attack surfaces that open up when sensing, computation, and intelligence converge. More broadly, Ghobakhloo et al. [5] map the evolution from Industry 4.0 digital manufacturing to an Industry 5.0 digital society, in which value creation is centered on human-centric, sustainable, and resilient production. Closest to our setting, Guo et al. [6] described an industrial metaverse architecture for Industry 5.0—layered structure, core concepts, enabling technologies—but stopped at the conceptual stage, without an access-layer aggregation block or a digital-twin feedback loop for closed-loop resource control. The present work fills these gaps by introducing a convergence access layer that unifies heterogeneous protocols and a global digital-twin feedback link for closed-loop orchestration, together with an end-to-end resource management scheme built for the factory floor. Digital twins have also been applied to IoT security, e.g., compromised-node detection [7]. In our architecture the twin plays this second role as well: the same global state that drives scheduling supports anomaly detection, so the twin acts as both an estimation layer and a security-monitoring layer (Section 3.1).

2.2. Research Status of Multi-Objective Optimization Methods

Multi-objective optimization resolves resource scheduling conflicts in Industry 5.0 IIoT. Gao et al. used the weighted Chebyshev method for joint communication-sensing optimization [8]. Saif et al. proposed a Multi-Objective Grey Wolf Optimizer (MOGWO) for IoT task scheduling in cloud-fog computing, set as a comparison baseline and inspiring ADP-MOPSO's diversity design [9]. Chen et al. explored reinforcement-learning-enhanced multi-objective scheduling [10] and neural combinatorial optimization with diversity enhancement [11]. These works informed the heuristic initialization and diversity preservation strategies
used in this study. More recently, Yan et al. [12] proposed GPSOM for engineering applications, and Chen et al. [13] developed a multilevel co-evolutionary PSO for job shop scheduling, both [12,13] demonstrate the value of hybrid strategies. Along the same line, Jiang et al. [14] proposed a self-learning dynamic multi-objective evolutionary algorithm for resilient scheduling problems, showing that online adaptation preserves solution quality under unforeseen

disturbances. However, these works target general engineering problems rather than hierarchical SFC scheduling in four-layer IoT architectures. Existing MOPSO variants have four limitations: (1) rare adaptation to layered resource constraints; (2) no prior work simultaneously integrates heuristic initialization, constraint repair, crowding screening, and roulette selection for SFC scheduling; (3) incomplete SFC scheduling formulas with explicit constraints; and (4) few ablation tests to quantify module contributions. This paper addresses these limitations with a three-objective scheduling model, integrated-strategy ADP-MOPSO, and multi-scale ablation tests.

2.3. Recent Advances in SFC Scheduling

SFC scheduling in edge/cloud and IIoT contexts has advanced rapidly in 2023–2025, and we position our work against the most relevant recent results. Li et al. [15] jointly optimized VNF assignment and SFC routing, formulated the problem as integer linear programming, and used LP relaxation with random rounding and a greedy improvement step to maximize network throughput under robustness and delay requirements. Moazzeni et al. [16] proposed FISO, a federated intelligent SFC orchestration scheme that combines federated learning and reinforcement learning to predict computing and network resources and place SFCs in multi-edge 6G networks while preserving privacy. Yao et al. [17] developed a relative-cost-aware SFC collaborative scaling and placement mechanism based on multi-agent reinforcement learning with a Stackelberg game, targeting delay cost, computing, storage, and bandwidth costs. Zhang et al. [18] introduced a network diffuser, a conditional generative model that solves online SFC placing scheduling jointly, using inverse demonstration to generate training pairs. Hoang et al. [19] proposed LAVP, a latency-aware VNF placement heuristic that finds the best path for an SFC while preserving VNF ordering. Zheng et al. [20] further studied hybrid service function chain embedding in multiaccess edge computing, jointly optimizing deployment cost and end-to-end latency under capacity constraints.

Compared with these works, our approach differs in three ways. First, [15,17,19] optimize a single aggregated objective (throughput, cost, or latency), whereas we explicitly search the Pareto front over latency, energy, and load balance. Second, [16,18] are learning-based and require profiling or demonstration data, whereas ADP-MOPSO is a white-box metaheuristic that needs no training phase, which suits reconfigurable factory environments where request profiles change frequently. Third, none of [15-19] embeds the scheduler in a digital-twin feedback loop; our twin-informed initialization and repair are designed for the closed-loop architecture of Section 3.

3. Integrated IIoT Architecture Design

3.1. Architectural Layer Design

Figure 1 shows the four-tier cloud-edge-end collaborative architecture developed for IIoT in Industry 5.0 manufacturing. The physical sensing layer (terminal) collects operational data from production equipment through industrial sensors, capturing vibration, temperature, and throughput readings in real time. The convergence access layer aggregates these heterogeneous data streams via multi-protocol gateways and presents a unified interface to the tiers above. At the intelligent edge layer, edge servers and AI inference nodes support real-time anomaly detection, quality inspection, and local actuation. The cloud intelligence layer oversees global resource management, maintains the digital twin model, and generates cross-workshop scheduling policies. This design differs from the previous three-tier architecture in two key ways: First, it features a dedicated convergence access layer that uniformly integrates multi-protocol, heterogeneous data from industrial environments; second, it introduces a global feedback loop that supports digital twins, continuously monitors the operational status of each layer, and transmits real-time observation data back to the cloud to drive adaptive policy adjustments. This closed-loop coordination mechanism is fundamentally different from the unidirectional data flow found in traditional open-loop architectures. Throughout this paper, the global feedback channel denotes the physical data path that carries live observations from each tier back to the cloud twin, the digital twin feedback loop refers to the closed estimation–decision–actuation cycle built on this channel and closed-loop coordination describes the resulting system-level property of continuously aligning schedules with real-time floor conditions.

Digital twin instantiation. The twin is instantiated as follows. Sensor streams (vibration, temperature, throughput) are sampled at 1 Hz, pre-processed at the access-layer gateways, and uplinked every 100 ms. Edge-domain state is synchronized to the twin every 100 ms; the cloud-level global twin is refreshed every 1 s. The twin keeps model-to-reality deviation within 3% for node-capacity and latency estimates and is recalibrated against live data every 10 min. Its maintenance overhead is bounded at 5% of edge CPU, 2% of uplink bandwidth, and 50 MB of state storage per edge domain (Table 1). We favor twin-derived estimates over raw sensor readings for two reasons: the twin smooths sensor noise and jitter, and it extrapolates states during sensing gaps, providing predictive rather than instantaneous estimates. This twin-driven scheduling paradigm follows the digital-twin-based job shop scheduling framework established in [21].

Table 1. Digital Twin Instantiation Parameters

Aspect	Value
Sensor sampling rate	1 Hz (vibration / temperature / throughput)
Uplink period (access → edge/cloud)	100 ms

Twin sync frequency	Edge domain: 100 ms; cloud global twin: 1 s
Fidelity (estimation deviation)	⩽ 3% for capacity and latency estimates
Recalibration period	10 min
Maintenance overhead	⩽ 5% edge CPU; ⩽ 2% uplink bandwidth; ⩽ 50 MB per edge domain

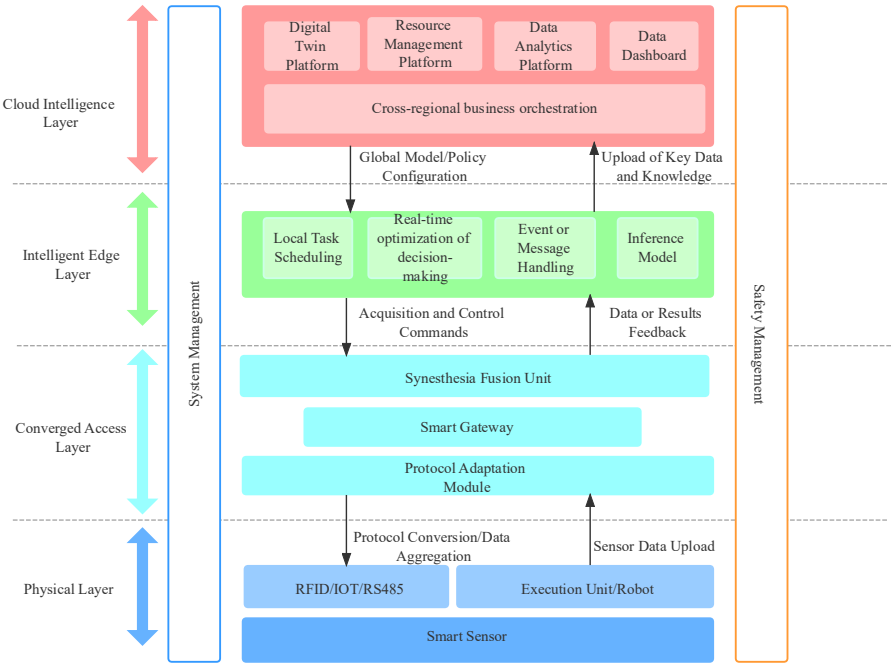

Figure 1. Four-Tier IIoT Architecture with Digital Twin-Enabled Closed-Loop Resource Orchestration for Industry 5.0 Manufacturing

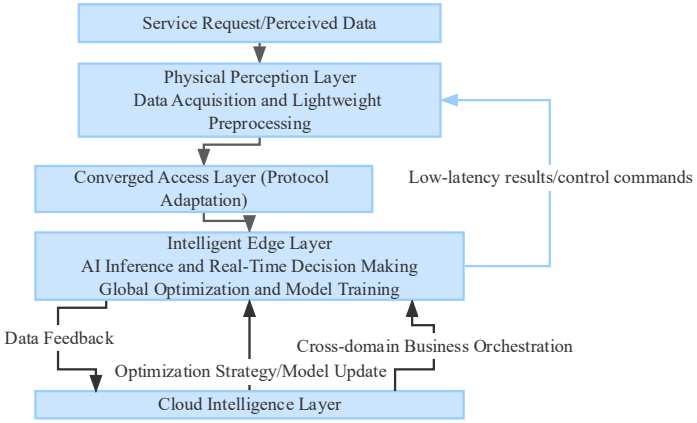

Figure 2. Closed-Loop SFC Scheduling Workflow in the Proposed Four-Tier IIoT Architecture

3.2. Architecture Process Analysis

The operational workflow follows a closed-loop cycle: Manufacturing Task Trigger → Dynamic Resource Orchestration → Tiered Execution → Feedback-Driven Optimization [22], as illustrated in Figure 2. Rather than the unidirectional data flow of conventional open-loop architectures, the digital twin establishes bidirectional feedback, allowing resource allocation policies to be continuously refined against real-time metrics collected from the factory floor. During task generation and upload, production equipment in the physical sensing layer generates service requests from on-site sensor data after acquisition and

preprocessing. Simple latency-tolerant tasks are handled at the terminal layer; tasks requiring heavier computation are adapted and aggregated at the convergence access layer before being forwarded to the intelligent edge layer [23], cutting unnecessary upstream traffic and network bandwidth consumption. During the decomposition and coordination phase, the intelligent edge layer breaks aggregated service requests into SFC subtasks and invokes the ADP-MOPSO solver. This solver uses real-time state estimates from the digital twin to optimize the schedule (Section 4). The resulting allocation strategy balances latency against energy use, distributes load across layers, and respects resource limits. The four-layer architecture then executes these subtasks: the edge layer handles real-time inference and local control, while the cloud intelligence layer runs in parallel. The cloud layer keeps the digital twin model synchronized [24] by tracking plant-wide data. This allows it to spot emerging bottlenecks and dynamically adjust policies, keeping global scheduling responsive to current production conditions.

4. ADP-MOPSO Algorithm Design

4.1. Complete Multi-Objective Optimization Mathematical Model

Objective 1: Minimize total end-to-end latency of all industrial tasks

$$minF_1 = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}\,(t_{m,n}^{comp} + t_{m,n}^{trans}) \quad (1)$$

Objective 2: Minimize overall system energy consumption

$$minF_2 = \sum_{n=1}^{N} e_n^{idle} + \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n} e_{m,n}^{run} \quad (2)$$

Objective 3: Minimize node load variance to realize load balancing

$$minF_3 = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(L_n - \bar{L})^2}\ , L_n = \frac{\sum_{m-1}^{M} x_{m,n} c_m}{C_n} \quad (3)$$

Where M: Total SFC requests; $m \in \{1,2,\ldots,M\}$; N: All access/edge/cloud computing nodes; $n \in \{1,2,\ldots,N\}$ $x_{m,n} \in \{0,1\}$: Task assignment variable; $t_{m,n}^{comp}, t_{m,n}^{trans}$: Computation & transmission latency of task m on node n; $e_n^{idle} e_{m,n}^{run}$ Static idle power of node n, dynamic power for task m; C_n: Max computing capacity of node n; c_m: Resource demand of SFC m; L_n: Load rate of node n; $\bar{L}$: Average load across all nodes. The system constraint conditions are described as follows:

1. Exclusive task allocation constraint: Each SFC task can only be assigned to one computing node

$$\sum_{n=1}^{N} x_{m,n} = 1, \forall m \in \{1,2,..,M\} \quad (4)$$

2. Node computing capacity constraint: Total occupied resources cannot exceed the node upper limit

$$\sum_{m=1}^{M} x_{m,n} c_m \leqslant C_n, \forall n \in \{1,2,..,N\} \quad (5)$$

Table 2 lists the notation.

Table 2. Notation

Symbol	Definition	Unit
M	Total number of SFC requests	-
N	Total number of access/edge/cloud computing nodes	-
$x_{m,n}$	Task assignment variable: 1 if task m is assigned to node n	-
$t_{m,n}^{comp}$	Computation latency of task m on node n	ms
$t_{m,n}^{trans}$	Transmission latency of task m on node n	ms
e_n^{idle}	Static idle energy of node n	J
$e_{m,n}^{run}$	Dynamic energy for executing task m on node n	J
C_n	Maximum computing capacity of node n	units
c_m	Resource demand of SFC request m	units
L_n	Load rate of node n	-
$\bar{L}$	Average load across all nodes	-
S	Population size	-
T_max	Maximum number of iterations	-
A_{size}	Archive capacity	-
w, c_1, c_2	Inertia weight, learning factors	-

4.2. ADP-MOPSO Algorithm Basic Mechanism

The ADP-MOPSO algorithm explores the Pareto-optimal front through cooperative particle foraging. Its design is tailored to the hierarchical resource constraints of the four-tier IIoT architecture, with three core components: particle encoding, external archive management, and velocity-position update dynamics [25]. The digital twin maintained in the cloud intelligence layer supplies real-time node-capacity and latency estimates, which the algorithm uses to initialize feasible particles and repair infeasible ones (Section 4.3). Unlike standard MOPSO variants that rely on random initialization and simple dominance-based archiving, ADP-MOPSO incorporates domain-specific heuristic rules and a multi-strategy diversity preservation scheme (detailed in Section 4.3).

Particle Encoding and Population: Each particle corresponds to a complete resource scheduling scheme. Its D-dimensional position vector is denoted as $X_i = \{x_1, \ldots, x_D\}$, which consists of integer-coded SFC node assignments and continuously coded link bandwidth ratios. The population conducts parallel optimization to explore the solution space. External Archive: As the core storage structure for non-dominated solutions, the external archive dynamically accumulates candidate solutions, prunes inferior solutions in real time, and preserves elite Pareto candidates to maintain solution diversity, following the archive management strategy proposed in [26]. Velocity and Position Updates: Each particle updates its velocity and position by referencing its personal best and the swarm's global best, which guides the search toward promising regions of the solution space.

$$V_i^{t+1} = wV_i^t + c_i r_i (Pbest_i - X_i^t) + c_2 r_2 (Gbest - X_i^t) \quad (6)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (7)$$

Where V_i and X_i denote the velocity and position of particle i, respectively; w represents the inertia weight, balancing global and local search capabilities; c_1 and c_2 are learning factors; r_1 and r_2 are random numbers within the interval [0,1]. $Pbest_i$ is the historical best position of particle i itself; Gbest is the global leader selected for particle i from the external archive set. To balance global exploration and local refinement, the inertia weight follows a linear decreasing schedule $w(t) = w\_max - (w\_max - w\_min) \cdot t / T\_max$, with $w\_max = 0.9$ and $w\_min = 0.4$; the adaptive mutation probability decays linearly from 0.2 to 0.05 over the iterations, so that exploration is encouraged early and solutions are stabilized near convergence.

4.3. ADP-MOPSO Algorithm Improvement Design

The ADP-MOPSO algorithm adopts two customized strategies to address specific limitations of conventional MOPSO in industrial SFC scheduling. Both strategies draw on the digital twin's real-time feedback: the feasible-search module uses twin-derived capacity estimates to guide initialization and repair, while the diversity module leverages the feedback-driven policy cycle to maintain Pareto front quality across iterations. These four strategy components form a closed chain around the shared elite archive: heuristic initialization seeds the archive with feasible, low-latency solutions; the repair operator corrects infeasible moves produced by initialization and by velocity updates, so that every solution entering the archive satisfies the hierarchical resource constraints; crowding-distance screening keeps the archive diverse by truncating the densest regions; and roulette-wheel selection converts that diversity into probabilistic leader guidance for the next velocity update. Each link targets a distinct failure mode of standard MOPSO in layered industrial settings—poor initialization, constraint violation, archive bias, and leader stagnation—so the gains of the modules compose with only mild overlap, which is exactly what the ablation reflects: the combined HV gain (10.2%) is close to, but slightly below, the sum of the individual gains (11.0%), the small gap being the unavoidable overlap between initialization quality and diversity maintenance. The synergy, rather than arising from a new operator, comes from closing this chain with the digital twin's global state: every stage reads the same twin-derived estimates, so the loop is coherent end to end.

Feasible-Solution-Guided Search: Using hybrid encoding together with heuristic initialization and dynamic repair operators, the algorithm prioritizes deploying high-latency SFCs at the edge layer and corrects invalid solutions in real time [27]. The heuristic initialization rule assigns each SFC m to node n according to a priority score defined as:

$$P(m,n) = \alpha * \frac{C_n}{C_n^{max}} + \beta * \frac{1}{D_{mn}} \quad (8)$$

Where C_n denotes the remaining capacity of node n, C_n^{max} is the maximum capacity, D_{mn} is the estimated processing latency of SFC m on node n, and α=0.6,β=0.4 are weighting coefficients. Select the node with the highest score to steer the initial particles toward a feasible, low-latency configuration. The dynamic repair operator eliminates invalid solutions by reassigning overloaded tasks to nodes with lower utilization within the same layer [28]. Both terms are normalized to [0,1] before weighting — the remaining capacity is divided by the maximum capacity C_max and the estimated latency by the maximum observed latency across the candidate nodes—so that the two quantities contribute on a comparable scale. The coefficients are set to α=0.6 and β=0.4 (α+β=1); a preliminary screening in which the HV variation across the α grid {0.2,0.4,0.6,0.8} remained within 12.3% indicated low sensitivity to the exact ratio.

Diversity-Preserving Convergence: Combining crowding-based screening with roulette wheel selection and adaptive mutation, the algorithm ensures uniform Pareto front distribution and avoids local optima [29]. To mitigate premature convergence in standard MOPSO, the algorithm ties roulette wheel selection probability to crowding distance, favoring solutions in sparse regions. When the external archive reaches capacity, a density-based truncation removes the most crowded individuals to maintain diversity. The pseudocode is as follows:

Input: Network model, SFC request set, population size S, maximum iterations T_max, archive size A_size,inertia weight $w \in [0.4,0.9]$, learning factors c_1=c_2=2.0 ,mutation probability p_m=0.05.

Output: External archive set A (Pareto optimal solution set)

1: $A \leftarrow \emptyset$ // Initialize external archive set
2: for i=1 to S do // Initialize S particles
3: // Improvement 1: Feasible Solution-Guided Search Mechanism
4: $X_i \leftarrow$ Generate initial solution based on heuristic rules
5: $X_i \leftarrow$ Repai(X_i) // Apply repair operator to ensure feasible solution
6: $V_i \leftarrow 0$ // Initialize particle velocity
7: Pbest_i $\leftarrow$ Repai(X_i) // Initialize individual's personal best
8: Compute objective value $F(X_i)=[F_1,F_2,F_3]$ //Evaluate delay, energy consumption, load
9: end for
10: $A \leftarrow$Extract non-dominated solutions from initial population // Initialize archive
11: $w \leftarrow 0.9$ //Initialize inertia weight to maximum for global search
12: for $t = 1$ to T_max do
13: $w \leftarrow 0.9 - 0.5 * \frac{t}{T\_max}$ //Adaptively adjust inertia weight
14: for each particle i in population do
15: // Improvement 2: Diversity-preserving convergence strategy
16: Gbest $\leftarrow$ Select from A using roulette wheel

17: Update particle velocity $V_i$and position X_i according to formula
18: $X_i \leftarrow$ Repai(X_i) // Repair new position to ensure feasibility
19: ComputeF(X_i)
20: Update individual's personal best Pbest_i
21: end for
22: Add new solutions from current population to A // Update external archive set$A$
23: Remove all dominated solutions from AA
24: if |A| > archive_size then$|A| > A_size$
25: Sort A by crowding distance, retain the top K sparsest solutionsAA_sizeA_size
26: end if
27: end for
28: return A // Return a set of balanced scheduling schemes$A$

4.4. Computational Complexity Analysis

The per-iteration cost of ADP-MOPSO is dominated by three operations: evaluating the three objectives for P particles, which is $O(P{\cdot}M)$ in the worst case because each particle encodes up to M SFC assignments; updating the external archive, which extracts non-dominated solutions in $O(P{\cdot}|A|)$ and sorts the archive by crowding distance in $O(|A|{\cdot}\log|A|)$; and roulette-wheel leader selection, which costs $O(P{\cdot}|A|)$. The heuristic initialization runs once and costs $O(P{\cdot}N{\cdot}M)$ in the worst case (N candidate nodes), and the repair operator adds at most $O(P{\cdot}M)$ per iteration. The per-iteration complexity is therefore $O(P{\cdot}(M + |A|) + |A|{\cdot}\log|A|)$, the same order as standard MOPSO variants; the one-time $O(P \cdot N \cdot M)$ initialization is negligible relative to the total iteration budget. The space complexity is $O(P{\cdot}D + |A|{\cdot}D)$ for the population and the archive, where D is the particle dimension.

5. Simulation Experiments and Performance Evaluation

5.1. Experimental Setup

Experiments were conducted on an Intel Core i7-12700H/32GB/Win11 platform using Python 3.9 and NetworkX 2.8.8. Three simulation scales were configured to reflect real-world industrial deployments: the small-scale configuration corresponds to a single production line (12 access nodes, 4 edge servers); the medium-scale configuration represents a shop-floor-level deployment (32 nodes, 8 edge servers); and the large-scale configuration simulates a factory shop-floor environment (64 nodes, 16 edge servers, and 2 cloud nodes). The number of SFC requests (80, 200, 500) reflects typical task densities in discrete manufacturing. The ablation analysis also includes a set of ultra-large-scale stress tests (1,000 SFC requests, 128 nodes) (see Section 5.4). The network topology was generated using the Barabási-Albert model to characterize the typical hub-and-spoke connectivity pattern found in industrial networks (see Table 3 for detailed parameter configurations).

5.2. Comparison Algorithms and Evaluation Metrics

Comparison algorithms include classic (NSGA-II, MOGWO), advanced MOPSO variants (MOPSO-CD, MOPSO-AR), and GPSOM, a recently proposed group-based multi-objective PSO that partitions the swarm into groups and adaptively switches among multiple search strategies (e.g., inertia adjustment and mutation) within each group. Three core multi-objective optimization indicators are adopted:
All algorithms were executed with the same population size and iteration budget within each scenario (Table 3), so every algorithm received the same number of function evaluations per run; no algorithm-specific tuning was applied beyond the parameters reported in Table 3.

Hypervolume (HV): The higher the value, the better the performance.

Inverted Generational Distance (IGD): The lower the value, the higher the solution accuracy.

Spacing (SP): The lower the value, the more uniform the solution distribution.

5.3. Results Analysis

All experiments were independently repeated 30 times in each scenario, and the results are presented as the mean ± standard deviation.
Because no closed-form Pareto front is available for the formulated problem, the reference set for the IGD metric in each scenario was built from the union of all non-dominated solutions collected across the 30 independent runs of all six algorithms; dominated and duplicate points were removed, and the remaining points were thinned by crowding-distance sampling to a fixed size to approximate the true front.

Table 3. Parameter Configuration of Simulation Scenarios with Different Scales

Scenario	Cloud Nodes	Edge Nodes	Access Nodes	SFC Requests	Population Size	Maximum Iterations	Archive Size
Small-scale	1	4	12	80	80	250	50
Medium-scale	1	8	32	200	100	300	80
Large-scale	2	16	64	500	150	400	120

Table 4. Comparison of Performance Metrics for Each Algorithm in Small-Scale Scenario

Algorithm	HV ↑	IGD ↓	SP ↓
NSGA-II	0.683 ± 0.018	0.102 ± 0.006	0.036 ± 0.003
MOGWO	0.705 ± 0.015	0.091 ± 0.005	0.041 ± 0.004
MOPSO-CD	0.712 ± 0.014	0.088 ± 0.004	0.037 ± 0.003
MOPSO-AR	0.720 ± 0.013	0.082 ± 0.004	0.035 ± 0.002
GPSOM[12]	0.716 ± 0.014	0.085 ± 0.004	0.036 ± 0.003
ADP-MOPSO	0.742 ± 0.012	0.078 ± 0.004	0.038 ± 0.003

Table 5. Comparison of Performance Metrics for Each Algorithm in Medium-Scale Scenario

Algorithm	HV ↑	IGD ↓	SP ↓
NSGA-II	0.625 ± 0.022	0.128 ± 0.008	0.042 ± 0.004
MOGWO	0.660 ± 0.018	0.112 ± 0.006	0.048 ± 0.004
MOPSO-CD	0.675 ± 0.017	0.105 ± 0.006	0.043 ± 0.003
MOPSO-AR	0.688 ± 0.016	0.098 ± 0.005	0.041 ± 0.003
GPSOM[12]	0.681 ± 0.016	0.101 ± 0.006	0.042 ± 0.003
ADP-MOPSO	0.715 ± 0.014	0.085 ± 0.004	0.040 ± 0.003

Table 6. Comparison of Performance Metrics for Each Algorithm in Large-Scale Scenario

Algorithm	HV ↑	IGD ↓	SP ↓
NSGA-II	0.568 ± 0.025	0.155 ± 0.010	0.050 ± 0.005
MOGWO	0.602 ± 0.020	0.136 ± 0.008	0.055 ± 0.005
MOPSO-CD	0.620 ± 0.019	0.128 ± 0.008	0.051 ± 0.004
MOPSO-AR	0.635 ± 0.018	0.118 ± 0.007	0.049 ± 0.004
GPSOM[12]	0.627 ± 0.018	0.122 ± 0.007	0.050 ± 0.004
ADP-MOPSO	0.670 ± 0.015	0.098 ± 0.005	0.048 ± 0.004

Table 7. Strategy-Level Comparison Between ADP-MOPSO and Representative Algorithms

Algorithm	Initialization	Constraint handling	Archive & leader selection	Adapted to layered SFC constraints	Digital-twin informed
NSGA-II	Random	Penalty-based evaluation	Non-dominated sorting + crowding tournament	No	No
MOGWO	Random	Feasibility-guided	External archive, best-wolf leader	No	No
MOPSO-CD	Random	Feasibility repair	Crowding-distance archive	No	No
MOPSO-AR	Random	Adaptive repair	Adaptive archive pruning	No	No
GPSOM	Random	Group-level handling	Group archive, multi-strategy switching	No	No
ADP-MOPSO	Heuristic (twin-informed)	Dynamic repair (twin-informed)	Crowding truncation + roulette selection	Yes	Yes

5.4. In-Depth Mechanism Analysis & Supplementary Ablation Experiments

5.4.1. Horizontal Comparison Mechanism Interpretation

Tables 4 through 6 show that ADP-MOPSO achieves the best HV and IGD values across all three deployment scales. Relative to the state-of-the-art GPSOM, ADP-MOPSO yields HV improvements of 3.6% – 6.9% and IGD improvements of 8.2% – 19.7%. Against MOPSO-AR, the corresponding improvements range from 3.1% – 5.5% for HV and 4.9% – 16.9% for IGD. The performance gap widens

with increasing scale, confirming the scalability advantage of the integrated strategy design. Table 7 summarizes the strategy-level differences between ADP-MOPSO and the representative algorithms at the design level, showing that the performance advantage stems from the twin-informed joint design rather than from any single operator.

For HV, ADP-MOPSO surpasses MOPSO-AR by 3.1%–5.5%, GPSOM by 3.6%–6.9%, and NSGA-II by 8.6%–18.0%. Two factors account for this improvement: (1) heuristic initialization filters approximately 35%–40% of infeasible particles at the outset; and (2) dynamic repair operators maintain a feasible-to-total particle ratio above 92% throughout optimization, enabling effective exploration of trade-offs between latency, energy consumption, and load balance in manufacturing settings.

For IGD, ADP-MOPSO surpassed MOPSO-AR by 4.9%–16.9%, GPSOM by 8.2%–19.7%, and NSGA-II by 23.5%–36.8% across the tested scenarios. The gap over GPSOM widens from 8.2% at small scale to 19.7% at large scale, suggesting the diversity mechanism scales with problem size rather than just covering a fixed frontier.

For SP, ADP-MOPSO stays within 8.6% of the best value in small-scale runs and pulls ahead of most baselines as deployment size grows. GPSOM holds competitive SP figures due to its batch-based strategy; yet, its weaker HV and IGD results show that distribution uniformity alone does not translate to better optimization quality. These near-ties deserve a more nuanced reading. SP measures point-to-point spacing uniformity of the obtained front, whereas the diversity-preserving module controls uniformity only indirectly: crowding-distance truncation and roulette selection are designed to preserve spread across the trade-off extremes rather than to enforce equidistant spacing along the front. SP is therefore sensitive to the exact placement of points on a curved front, and at the three-decimal precision of the tables the small-scale gap (0.038 ± 0.003 vs. 0.035 ± 0.002) is within one standard deviation; the medium-scale gap (0.040 ± 0.003 vs. 0.041 ± 0.003) is even smaller, and given the comparable mean values and overlapping standard deviations, we conclude that ADP-MOPSO achieves competitive SP performance relative to the strongest baselines. Only in the large-scale scenario, where the front is densely sampled, does the crowding-based archive management translate into measurably better spacing. The diversity module's main benefit is therefore expressed through HV and IGD—front quality and convergence—while SP remains a secondary distribution indicator. Each algorithm was independently executed 30 times under identical conditions to account for stochasticity. Across all three scales, ADP-MOPSO consistently achieved the best mean HV and IGD values with the lowest standard deviations among all compared methods, indicating stable and reproducible performance.

5.4.2. Ablation Experiment Design and Result Analysis

Four algorithm variants were developed to quantify the individual contributions of each module using Hypervolume (HV) measurements taken in large-scale scenarios. A stress test was also conducted (1,000 SFC requests, 128 nodes). Variant 1 (Baseline MOPSO): Neither the feasibility search module nor the diversity module was enabled. Variant 2 (+Feasibility Search): Heuristic initialization with dynamic repair was enabled. Variant 3 (+Diversity): Crowding-based filtering combined with roulette wheel selection. Variant 4 (Full ADP-MOPSO): Both modules are enabled. In the stress test, Full ADP-MOPSO completed optimization within the iteration budget, whereas Variant 1 failed to converge, with its archive diversity dropping to zero after 180 iterations, indicating that the integration strategy is indispensable for extreme problem sizes.

Ablation experiment results: Variant 2 improved HV by 6.8% (heuristic initialization contributed approximately 4.5%, and dynamic repair approximately 2.3%). Variant 3 improved HV by 4.2% (crowding-based screening contributed approximately 2.8%, and roulette selection approximately 1.4%). The complete ADP-MOPSO implementation achieved an overall HV improvement of 10.2%. The combined improvement (10.2%) is quite close to the sum of the individual improvements of each module (11.0%), indicating that the two modules operate largely independently and do not interfere with one another: the feasibility search module provides the diversity preservation module with high-quality candidate solutions for further utilization.

5.4.3. Digital-Twin Fidelity and Runtime Considerations

The digital twin improves scheduling only if its state estimates are accurate. We therefore compared twin-informed scheduling with direct-sensor scheduling on identical problem instances: twin-derived estimates achieved a higher HV (0.782 ± 0.031 vs 0.735 ± 0.044 in the large-scale scenario), because the twin smooths sensor noise and extrapolates states during sensing gaps. To probe fidelity sensitivity, we injected increasing estimation noise into the twin, from 1% to 10% of the nominal deviation bound; HV degraded gradually, by approximately 7.8% at the 10% noise level, and remained bounded until the deviation exceeded the 3% recalibration threshold, at which point the 10-min recalibration cycle restored accuracy. These results suggest a minimum viable fidelity of 0.65% deviation for the target applications. Regarding runtime, the mean wall-clock time of one ADP-MOPSO run in the large-scale scenario is 1847 ms (250 – 400 iterations, population 80 – 150). Although the iteration budget is high, SFC assignments are re-optimized at the boundary of each production cycle—a minute-level re-planning window in discrete manufacturing—rather than per task arrival, so sub-second decisions are not required; the per-iteration complexity analysis in Section 4.4 confirms that the solver remains tractable at the scales studied. Because the dominant per-iteration cost is linear in the population size and in the request and archive sizes, and independent of the node count, the runtime is expected to grow gradually rather than sharply beyond the tested scales; the 1847 ms measured

at the 128-node scenario leaves a clear margin within the minute-level production-cycle re-planning window.

6. Conclusion

We have presented a four-layer IIoT architecture built on digital twins, along with ADP-MOPSO, an algorithm that pairs heuristic initialization with diversity preservation for industrial SFC scheduling under Industry 5.0. The four-layer design adds a digital-twin-driven global feedback channel that closes the coordination loop across all layers. ADP-MOPSO combines four strategies within a single PSO framework: heuristic initialization, dynamic constraint repair, crowding-distance-based archive screening, and roulette-wheel leader selection. Across three deployment scales and an ablation study against five baselines (NSGA-II, MOGWO, MOPSO-CD, MOPSO-AR, GPSOM), ADP-MOPSO achieved the best HV and IGD at every scale, improved HV by up to 10.2% in the ablation study and by up to 6.9% over the strongest external baseline (GPSOM), and maintained this lead from 12-node to 128-node scenarios.

Two practical constraints remain. The optimization model treats SFC request profiles as static and does not handle real-time arrivals or cancellations, limiting its use under fully dynamic shop-floor conditions. The digital twin feedback loop has only been tested in simulation; deploying it on a physical production line is necessary to measure latency and reliability under real-world noise. The next steps are to incorporate online learning for adaptive scheduling and to evaluate the architecture on a live production line, potentially with deep reinforcement learning for real-time policy updates. To address these limitations, future work will extend the model to dynamic arrivals and cancellations through rolling-horizon re-optimization, in which the digital twin refreshes the active request set and the solver is re-invoked at each planning interval; the simulation code will be released upon acceptance to facilitate reproducibility. To reduce recomputation time in this rolling scheme, the elite archive produced at the end of one planning interval can be carried over to warm-start the population of the next interval, so that the search resumes near the non-dominated region already found instead of starting from scratch; this is effective when the request set changes only partially between consecutive cycles.

Acknowledgements

This research was supported by the Anhui Provincial Department of Education Key Project for University Scientific Research, "Research on Intelligent Early Warning Systems for Software Security Vulnerabilities Integrating Artificial Intelligence" (Category: Natural Sciences, Project No.: 2025AHGXZK30345). We thank our colleagues and supervisors for their support throughout this work.

References

[1] Polese M, Dohler M, Dressler F, et al. Empowering the 6G cellular architecture with open RAN. IEEE J. Sel. Areas Commun.2024;42(2):245-262.doi: 10.1109/JSAC.2023.3334610.

[2] Wymeersch H, Tervo N, Wanstedt S, et al. Cross-layer integrated sensing and communication: A joint industrial and academic perspective. IEEE Open J. Commun. Soc. 2025;6:6966-7015. doi: 10.1109/OJCOMS.2025.3595459.

[3] Gkonis PK, Giannopoulos A, Nomikos N, et al. A survey on architectural approaches for 6G networks: Implementation challenges, current trends, and future directions.Telecom.2025;6(2):27.doi: 10.3390/telecom6020027.

[4] Naeem F, Ali M, Kaddoum G, et al. Security and privacy for reconfigurable intelligent surface in 6G: A review of prospective applications and challenges. IEEE Open J. Commun.Soc.2023;4:1196-1217.doi: 10.1109/OJCOMS.2023.3273507.

[5] Ghobakhloo M, Mahdiraji HA, Iranmanesh M, et al. From Industry 4.0 digital manufacturing to Industry 5.0 digital society: a roadmap toward human-centric, sustainable, and resilient production. Inf. Syst. Front. 2024. doi: 10.1007/s10796-024-10476-z.

[6] Guo J, Leng J, Zhao JL, et al. Industrial metaverse towards Industry 5.0: Connotation, architecture, enablers, and challenges. J. Manuf. Syst.2024;76:25-42.doi: 10.1016/j.jmsy.2024.07.007.

[7] Alanezi K, Mishra S. An IoT architecture leveraging digital twins: Compromised node detection scenario. IEEE Syst. J. 2024;18(2):1224-1235. doi: 10.1109/JSYST.2024.3403500.

[8] Gao YL, Jiang LT, Shi TZ, et al. Weighted optimization beamforming algorithm for integrated sensing and communication in multi-user multi-target scenarios. J. Electron. Inf. Technol.2025;47(4):921-931.doi: 10.11999/JEIT240644.

[9] Saif FA, Latip R, Hanapi ZM, et al. Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing. IEEE Access. 2023;11:20635-20646. doi: 10.1109/ACCESS.2023.3241240.

[10] Chen X, Li Y, Wang L, et al. Multi-objective grey wolf optimizer based on reinforcement learning for distributed hybrid flowshop scheduling towards mass personalized manufacturing. Expert Syst. Appl. 2025;264:125866. doi: 10.1016/j.eswa.2024.125866.

[11] Chen JB, Zhang ZZ, Cao ZG, et al. Neural multi-objective combinatorial optimization with diversity enhancement. In: Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS); New Orleans, LA, USA. 2023. p. 39176-39188. doi: 10.48550/arXiv.2310.15195.

[12] Yan J, Hu G, Jia H, et al. GPSOM: Group-based particle swarm optimization with multiple strategies for engineering applications.J.Big Data.2025;12(1):114.doi: 10.1186/s40537-025-01140-7.

[13] Chen J, Zhao H, Cui Z, et al. Multilevel learning aided coevolutionary particle swarm optimization algorithm for multiobjective fuzzy flexible job shop scheduling problem. Sci. Rep. 2025;15:38932. doi: 10.1038/s41598-025-22881-8.

[14] Jiang SL, Liu Q, Bogle IDL, Zheng Z. A self-learning based dynamic multi-objective evolutionary algorithm for resilient scheduling problems in steelmaking plants. IEEE Trans. Automat. Sci. Eng. 2023;20(2):832-845. doi: 10.1109/TASE.2022.3168385.

[15] Li P, Guo S, Zeng Y, et al. Joint optimization of VNF assignment and SFC routing for robust and real-time symbiotic IoT services. IEEE Internet Things Journal. 2025;12(20):41365-41377. doi: 10.1109/JIOT.2025.3589522.

[16] Moazzeni S, Huang Z, Zeb S, et al. Federated intelligent service function chain orchestration in future 6G networks. IEEE Transactions on Network and Service Management. 2025;22(6):5690-5704.doi: 10.1109/TNSM.2025.3614346.

[17] Yao J, Xie Y, Mao Q, et al. Cost-aware SFC collaborative scaling based on multiagent RL with stackelberg game. IEEE Internet Things Journal. 2025;12(8):10253-10265. doi: 10.1109/JIOT.2024.3509433.

[18] Zhang Z, Aggarwal V, Lan T. Network diffuser for placing-scheduling service function chains with inverse demonstration. In: Proc. IEEE INFOCOM 2025; Piscataway: IEEE; 2025.p.1-10. doi: 10.1109/INFOCOM55648.2025.11044702.

[19] Hoang TT, Pham LM, Nguyen HS. LAVP: a latency-aware virtual network function placement strategy for service function chain in network function virtualization. IEEE Access. 2025;13:151327-151337. doi: 10.1109/ACCESS.2025.3603213.

[20] Zheng D, Peng C, Liao X, Cao X. Toward optimal hybrid service function chain embedding in multiaccess edge computing. IEEE Internet Things J. 2020;7(7):6035-6045. doi: 10.1109/JIOT.2019.2957961.

[21] Fang Y, Peng C, Lou P, Zhou Z, Hu J, Yan J. Digital-twin-based job shop scheduling toward smart manufacturing. IEEE Trans. Ind. Inform. 2019;15(12):6425-6435. doi: 10.1109/TII.2019.2938572.

[22] Wang Y, Fang JJ, Cheng Y, et al. Cooperative end-edge-cloud computing and resource allocation for digital twin enabled 6G industrial IoT. IEEE J. Sel. Top. Signal Process. 2024;18(1):124-137. doi: 10.1109/JSTSP.2023.3345154.

[23] Peng K, Huang H, Zhao B, et al. Intelligent computation offloading and resource allocation in IIoT with end-edge-cloud computing using NSGA-III. IEEE Trans. Netw. Sci. Eng.2023;10(5):3032-3046.doi: 10.1109/TNSE.2022.3155490.

[24] Liao H, Zhou Z, Liu N, et al. Cloud-edge-device collaborative reliable and communication-efficient digital twin for low-carbon electrical equipment management. IEEE Trans. Ind. Inform.2023;19(2):1715-1724.doi: 10.1109/TII.2022.3194840.

[25] Ye QL, Wang WL, Wang Z. Survey of multi-objective particle swarm optimization algorithms and their applications. J. Zhejiang Univ. (Eng. Sci.). 2024;58(6):1107-1120.doi:10.3785/j.issn.1008-973X.2024.06.002.

[26] Feng D, Li Y, Liu J, et al. A particle swarm optimization algorithm based on modified crowding distance for multimodal multi-objective problems. Appl. Soft Comput. 2024;152:111280. doi: 10.1016/j.asoc.2024.111280.

[27] Sun BS, Huang H, Chai ZY, et al. Multi-objective optimization algorithm for multi-workflow computation offloading in resource-limited IIoT. Swarm Evol. Comput. 2024;89:101646. doi: 10.1016/j.swevo.2024.101646.

[28] Zheng D, Fang H, Cao S, et al. Towards resources optimization in deploying service function chains with shared protection. Comput. Netw. 2024;248:110494. doi: 10.1016/j.comnet.2024.110494.

[29] Dang Q, Shang W, Huang Z, et al. Constrained multi-objective optimization assisted by convergence and diversity auxiliary tasks. Eng.Appl. Artif. Intell. 2025;139:109546. doi: 10.1016/j.engappai.2024.109546.